

8-28-00

A

Customer No. 20350
TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
(415) 576-0200

Attorney Docket No. 20533-000210US

"Express Mail" Label No. EL496155546US

Date of Deposit: 8-23-00

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above, addressed to:

Assistant Commissioner for Patents
Washington, D.C. 20231

By: Ron Anton

ASSISTANT COMMISSIONER FOR PATENTS
BOX PATENT APPLICATION
Washington, D.C. 20231

Submitted herewith for filing under 37 CFR 1.53(b) is the
 patent application of
 continuation patent application of
 divisional patent application of
 continuation-in-part patent application of

Inventor(s)/Applicant Identifier: Richard David Day, Eric Sven-Johan Swildens, Ajit Gupta

For: METHOD FOR OPERATING AN INTEGRATED POINT OF PRESENCE SERVER NETWORK

This application claims priority from each of the following Application Nos./filing dates:
60/166,906/November 22, 1999

the disclosure(s) of which is (are) incorporated by reference.

Please amend this application by adding the following before the first sentence: "This application is a continuation continuation-in-part of and claims the benefit of U.S. Provisional Application No. 60/_____, filed _____, the disclosure of which is incorporated by reference."

Enclosed are:

- 21 page(s) of specification
- 7 page(s) of claims
- 1 page of Abstract
- 7 sheet(s) of formal informal drawing(s).
- An assignment of the invention to Speedera Networks, Inc. including Recordation Form Cover Sheet.
- A signed unsigned Declaration & Power of Attorney
- A signed unsigned Declaration.
- A Power of Attorney by Assignee with Certificate Under 37 CFR Section 3.73(b).
- A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27 is enclosed was filed in the prior application and small entity status is still proper and desired.
- A certified copy of a _____ application.
- Information Disclosure Statement under 37 CFR 1.97 (including Form PTO-1449 and refs).
- A petition to extend time to respond in the parent application.
- Notification of change of power of attorney correspondence address filed in prior application.
- Postcard

	(Col. 1)	(Col. 2)
FOR:	NO. FILED	NO. EXTRA
BASIC FEE		
TOTAL CLAIMS	20 - 20	= *0
INDEP. CLAIMS	3- 3	= *0
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENTED		

* If the difference in Col. 1 is less than 0, enter "0" in Col. 2.

SMALL ENTITY	
RATE	FEE
	\$345.00
x \$9.00 =	\$0.00
x \$39.00 =	\$0.00
+ \$130.00 =	
TOTAL	\$345.00

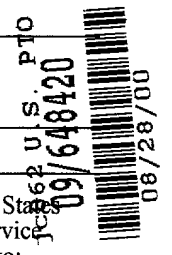
OTHER THAN SMALL ENTITY	
RATE	FEE
	\$690.00
x \$18.00 =	
x \$78.00 =	
+ \$260.00 =	
TOTAL	

Please charge Deposit Account No. 20-1430 as follows:

- Filing fee \$ 345.00
- Recordation Fee \$ 40.00
- Any additional fees associated with this paper or during the pendency of this application.
- The issue fee set in 37 CFR 1.18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b)

08/28/00

U.S. PATENT AND TRADEMARK OFFICE

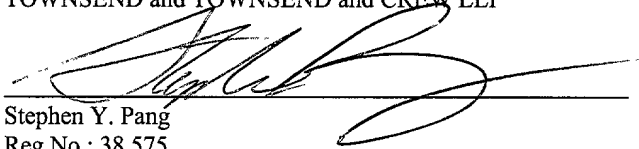


[] A check for \$_____ is enclosed.
2 extra copies of this sheet are enclosed.

Telephone:
(650) 326-2400

Facsimile:
(650) 326-2422

Respectfully submitted,
TOWNSEND and TOWNSEND and CREW LLP



Stephen Y. Pang
Reg No.: 38,575
Attorneys for Applicant

PA 3088809 v1

PA 3088809 v1

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) & 1.27(c)) - SMALL BUSINESS CONCERN

Applicant or Patentee: Richard David Day et al.
Application or Patent No.: _____
Filed or Issued: Herewith
Title: METHOD FOR OPERATING AN INTEGRATED POINT OF PRESENCE SERVER NETWORK

I hereby declare that I am:

- the owner of the small business concern identified below;
- an official of the small business concern empowered to act on behalf of the concern identified below.

Name of Small Business Concern: Speedera Networks, Inc.
Address of Small Business Concern: 4800 Great America Parkway
Santa Clara, CA 95054

I hereby declare that the above-identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention, entitled METHOD FOR OPERATING AN INTEGRATED POINT OF PRESENCE SERVER NETWORK by inventor(s) Richard David Day, Eric Sven-Johan Swidens, Ajit Gupta described in:

- the specification filed herewith;
- Application No. _____, filed _____;
- Patent No. _____, issued _____.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern that would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

*NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

Name: _____
Address: _____
 Individual Small Business Concern Nonprofit Organization

Name: _____
Address: _____
 Individual Small Business Concern Nonprofit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Name of Person Signing: AJIT K. GUPTA
Title of Person if Other than Owner: President and CEO
Address of Person Signing: 4800 Great America Parkway
Santa Clara, CA 95054

Signature  Date 08/21/00

**METHOD FOR OPERATING AN INTEGRATED
POINT OF PRESENCE SERVER NETWORK**

Inventors:

Richard David Day, a citizen of United States, residing at 912 Rich Avenue #3, Mountain View, CA 94040;

Eric Sven-Johnan Swildens, a citizen of United States, residing at 723 Tiana Lane, Mountain View, CA 94041; and

Ajit Gupta, a citizen of India, residing at 33635 Quail Run Road, Fremont CA 94555

Assignee:

Speedera Networks, Inc.
4800 Great America Parkway
Santa Clara, CA 95054

Entity: Small business concern

METHOD FOR OPERATING AN INTEGRATED POINT OF PRESENCE SERVER NETWORK

CROSS-REFERENCES TO RELATED APPLICATIONS

5 The present invention disclosure claims priority to and incorporates by reference for all purposes, Provisional U.S. Patent Application Number 60/166,906, Attorney Docket Number 22119-04581 (Case 4851 US), filed November 22, 1999, titled “WAN Traffic Management System and Method.”

BACKGROUND OF THE INVENTION

10 The present invention relates to world wide area networking. More particularly, the invention provides a technique including a method and system for using a global traffic management system coupled to a plurality of content servers for a service fee. But it would be recognized that the invention has a much broader range of
15 applicability. For example, the invention can also be applied to operation on a private network, and the like.

The Internet is a world wide “super-network” which connects together millions of individual computer networks and computers. The Internet is generally not a single entity. It is an extremely diffuse and complex system over where no single entity
20 has complete authority or control. Although the Internet is widely know for one of its ways of presenting information through the World Wide Web (herein “Web”), there are many other services currently available based upon the general Internet protocols and infrastructure.

The Web is often easy to use for people inexperienced with computers.
25 Information on the Web often is presented on “pages” of graphics and text that contain “links” to other pages either within the same set of data files (i.e., Web site) or within data files located on other computer networks. Users often access information on the Web using a “browser” program such as one made by Netscape Communications Corporation (now America Online, Inc.) of Mountain View, California or Explorer[™] from Microsoft
30 Corporation of Redmond, Washington. Browser programs can process information from Web sites and display the information using graphics, text, sound, and animation. Accordingly, the Web has become a popular medium for advertising goods and services directly to consumers.

As time progressed, usage of the Internet has exploded. There are literally millions of users on the Internet. Usage of the Internet is increasing daily and will eventually be in the billions of users. As usage increases so does traffic on the Internet. Traffic generally refers to the transfer of information from a Web site at a server computer to a user at a client computer. The traffic generally travels through the world wide network of computers using a packetized communication protocol, such as TCP/IP. Tiny packets of information travel from the server computer through the network to the client computer. Like automobiles during "rush hour" on Highway 101 in Silicon Valley, the tiny packets of information traveling through the Internet become congested. Here, traffic jams that cause a delay in the information from the server to the client occur during high usage hours on the Internet. These traffic jams lead to long wait times at the client location. Here, a user of the client computer may wait for a long time for a graphical object to load onto his/her computer.

From the above, it is seen that an improved way to transfer information over a network is highly desirable.

SUMMARY OF THE INVENTION

The present invention relates to a computer server network. More particularly, the present invention relates to an integrated load balancing and content delivery computer network.

Embodiments of the present invention include a number of Linux machines (on other operating machines) running software available from the current assignee. This software currently includes eight separate software components that are delivered as a single product. When deployed across a large number of machines, the software creates a network of point of presence (POP) computers that provides a solution for integrated content hosting and traffic management.

Customers using the POP network, according to one embodiment, can store content such as HTML, images, video, sound, software, or the like on the network for fast and highly available access by clients (end users). The network also provides load balancing and high availability for servers outside the network. Customers with generated content, such as search engines, auctions and shopping carts, can use the latter feature to add their own content servers to the network.

The computer network may be monitored using a standard web browser in some embodiments. In particular, HTML interfaces are provided that display the computer networks current status as well as historical statistics.

According to one embodiment of the invention a method for operating a network of point of presence servers sharing a hostname is disclosed. The method includes a user making a request for a web page with a domain name the POP network is authoritative for. The user's computer sends a request for name resolution of the domain name they are requesting content from to the configured local DNS server (LDNS) for that users computer. The LDNS is typically configured in close proximity to the user's computer. The LDNS resolves the IP address of the domain name the user's computer has queried for by sending a query to the Authoritative DNS server (ADNS) for that domain name.

The POP network DNS servers (SPDNS) is designated as an ADNS for all domains the POP network provides service for. On being asked to resolve the domain name from the LDNS on behalf of the user, the SPDNS first determines from the plurality of POP network servers and customer web servers the subset of servers addressable by the requested domain name. In this embodiment, addressable servers are servers that have been configured by either the customer or the POP network to satisfy user requests for content within the requested domain name. These servers have been defined in the POP network configuration as such.

From this subset of servers, the POP network selects the optimal server/servers for the user based on rules defined in the POP network configuration. These rules define how criteria such as POP/customer server health, latency from POP/customer servers to the user making the request, packet loss experienced at each POP, cost associated with each POP/customer server, and the like. These factors are then combined to define the optimal server for a given user making a request for a given domain name. In addition, persistence is defined in embodiments of the present invention for a domain name so that the user is returned to the same POP/customer server on subsequent requests within a definable time. These rules for defining the optimal server are created on a per domain name basis allowing each domain name to be configured independently. The next step includes the SPDNS returning the IP address of the optimal servers to the LDNS making the request, and the LDNS resolving the domain name for the user computer making the request for content with the IPs provided by the SPDNS.

In the present embodiment, the POP network servers are defined in the POP network configuration to be addressable by a requested domain names that represent a customer's static or cacheable content, a static domain name. The customer's servers are defined as addressable for domain names that represent dynamic or non-cacheable content, a dynamic domain name. Resolution for either domain name type is performed by the same set of servers controlled and operated by the POP network, the SPDNS. In addition, both domain name types leverage the same infrastructure for determining network and service load as well as configuration.

This technique allows the customer to organize their web site by functions where functions are mapped into domain names. The functions have requirements that define what is needed by a server to satisfy the function. Some functions may only be limited by the ability to access a collection of static content while others require a proprietary application and access to a customer's database. The domain names are then mapped into resources or servers that can be addressed by the domain name and that satisfy requests for the function that domain name has been defined to support. This method of organization allows individual elements of a web page to be retrieved from different resources. Further, such embodiments provides the SPDNS the control to direct users to an available and optimal server for a given resource. This greatly improves the user experience by reducing connection time, download time, errors, and the like.

According to one aspect of the invention a method for operating a network of point of presence servers sharing a hostname is disclosed. The method includes receiving a request from a user for a web page at a first web address, the first web address including the hostname, determining traffic loads of a plurality of customer web servers, each of the customer web servers storing the web page, and determining a customer web server from the plurality of customer web servers that is appropriate for the request, the customer web server having a traffic load and/or latency lower than traffic loads of remaining customer web servers from the plurality of customer web servers. The technique also includes determining an IP address of the customer web server, and providing the IP address of the customer web server to the user. The method may thereafter include receiving a request from the user for static content on the web page at a second web address, the second web address including the hostname, and determining latency of point of presence servers in the network of point of presence servers. Determining the point of presence server from the network of point of presence servers that is appropriate for the request, the point of presence server having a latency and/or

server load lower than latency of remaining point of presence servers from the network of point of presence servers, retrieving the static content from the point of presence server, and providing the static content to the user may also be performed.

According to another aspect of the invention, a method for operating a network of point of presence servers is disclosed. The method includes receiving a first request from a client DNS server to resolve a first domain name, the client DNS server receiving a request from a user of a web page address that includes the first domain name, and determining load measurements of a plurality of customer web servers, each of the customer web servers addressable by the first domain name, and each of the customer web servers configured to service the request from the user. The technique may also include determining a customer web server from the plurality of customer web servers, the customer web server having load measurements lower than load measurements (also including latency and other factors) of other customer web servers from the plurality of customer web servers latency, determining an IP address of the customer web server, and providing the IP address of the customer web server to the client DNS server.

Thereafter, the method typically includes receiving a second request from the client DNS server to resolve a second domain name, the client DNS server receiving a request from the user of a uniform resource locator that includes the second domain name, and sending the request to the authoritative DNS for this domain name (SPDNS) who determines performance metric measurement of point of presence servers in the network of point of presence servers, each of the point of presence servers addressable by the second domain name. Further, determining a point of presence server from the network of point of presence servers, the point of presence server having performance metric measurements lower than performance metric measurements of other point of presence servers from the network of point of presence servers, providing the IP address of the point of presence server to the client DNS server, retrieving data from the point of presence server in response to the uniform resource locator, and providing the data to the user are included.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to more fully understand the present invention, reference is made to the accompanying drawings. Understanding that these drawings are not to be considered limitations in the scope of the invention, the presently preferred embodiments and the presently understood best mode of the invention are described with additional detail through use of the accompanying drawings in which:

Fig. 1 is an overview block diagram of an embodiment of the present invention;

Fig. 2 illustrates a block diagram of a typical server machine according to an embodiment of the present invention;

5 Figs. 3A-3C illustrate a flow chart according to an embodiment of the present invention;

Fig. 4 illustrates a flow chart according to another embodiment of the present invention; and

10 Fig. 5 illustrates a more detailed block diagram according to one embodiment.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Fig. 1 is an overview block diagram of an embodiment of the present invention. Fig. 1 illustrates a series of “point of presence” (POP) servers 10, 20, and 30, customer web servers (POP servers) 40 and 45, a monitoring POP server 50, and an ISP cluster 60. In Fig. 1, the servers 10-50, and ISP 60 are typically coupled via a computer network 70, such as the Internet. In other embodiments, other types of computer networks may include other wide area networks, local area networks, such as intranets, and the like.

20 In the present embodiment, POP servers 10-30 are typically distributed around the world and together they form a POP network. In a typical embodiment, each POP server will include a plurality of dedicated servers, providing functionality as will be described below. For example, POP server 20 includes servers 65, 80-100. In this example, server 65 operates as a probing server according to NetProbe software described below; server 80 and 90 operate as a cache according to WebCache software described below; and server 100 operates as a content server according to WebServer software described below. As illustrated in Fig. 1, server 10 and 20 may include a server 110 and 120 that includes Domain Name Server (DNS) software, described below.

30 In the present embodiment, NetProbe server software performs probes across computer network 70 to determine server load and availability (metrics), to perform content checks to ensure servers are delivering correct content, to determine latency between customer web sites and POP servers, and the like. As will be described further below, NetProbe software may operate continuously and typically sends results to POP servers running NameServer software, such as POP server 10 and 30.

In embodiments of the present invention, measurements of server metrics may include response times for HTTP service, FTP service, customer web server CPU load, customer web server memory load and/or utilization, or the like. In other embodiment, other types of measurements are contemplated.

5 In this embodiment, the WebCache server software responds to requests for static web content from clients, such as web browsers from memory. In this example, if the requested content does not exist in server memory, the WebCache software will generate a request to a WebServer. When WebServer sends the requested content, the WebCache software caches the content in memory.

10 As described above, the WebServer software responds to requests for content from WebCache servers. If the content requested does not exist on disk, the WebServer software will generate a request to a web server, that is known to include the content, for example, web server 130. Once the content is provided by web server 130, the WebServer software stores the content to disk and also returns the content to the
 15 WebCache. In this embodiment, this server only gets request for WebCache "misses" so it does not require caching functionality.

 In the present embodiment, DNS software performs a mapping from domain name to IP address. As an example, the DNS software is queried to resolve a hostname, or the like, and returns an IP address of a web server that has the bandwidth to
 20 respond to the request. Further, in response to a query, the DNS software returns an IP address of a web server that can quickly provide web content.

 In Fig. 1, the dedicated servers are typically coupled to the computer network 70 via a router. In alternative embodiments, fewer number of physical servers may be used to implement functionality of servers 65, 80-100, or the like. For example, a
 25 physical server may have multiple software packages operating upon it at one time. As an example, WebCache software and WebServer software may operate as two different processes on a single server. Generally, each POP server will include NetProbe, WebCache, and WebServer software. In one embodiment, only a limited number of POP servers will include DNS software, however, in other embodiments, the number of
 30 servers with DNS software may be increased.

 As shown in Fig. 1, customer web servers 40 and 45 typically include web servers 130 and 135 and servers 140 and 145. In this embodiment, web servers 130 and 135 provide static content, such as text and graphics, further servers 140 and 145 provide non-static content, or content that changes in response to a user query, or the like. In

some embodiments, servers 140 and 145 may be search engines, databases, or the like. Typically, customer POP web servers 40 and 45 are mirrors of each other, and are designed to improve the perceived responsiveness of a web site with respect to a user.

ISP 60 is illustrated including users 150-160, and a client DNS server 170.

5 In this embodiment, ISP 60 may be a conventional ISP such as Earthlink, AOL, or the like. Accordingly, users 150 and 160 may be home users using dial-up lines, cable modems, DSL, or the like, to connect with ISP 60. As illustrated, client DNS server 170 is typically provided within ISP 60. In the present embodiment, client DNS server 170 is typically used to resolve a host name into an IP address. If client DNS server 170 does
10 not know the IP address, client DNS server 170 may pass the request to other DNS servers, such as DNS servers 110-120, or the like.

In the example in Fig. 1, a monitoring POP server 50 may include a plurality of dedicated servers, providing functionality described herein. For example, monitoring POP server 50 includes servers 180-210. In this example, server 180 operates
15 as an monitoring display server according to NetView software described below; server 190 operates as an administration server according to tools, together termed AdminTool software, described below; and servers 200 and 210 operate as logging servers according to LogServer software described below. As illustrated in Fig. 1, POP server 10 and 20 may include a server 110 and 120 that includes Domain Name Server (DNS) software, described below.
20

In the present embodiment, NetView software provides an HTML interface to current and historical statistics for customers and network administrators. The statistics about the POP network, and the like are typically derived from servers running LogServer software, such as servers 200 and 210, described below. In this
25 embodiment, NetView software includes web server CGI programs to provide the HTML output. The NetView software also provides an interface that allows customers to flush content from the POP network as they update the content on their web servers.

LogServer software, in this embodiment, receives log file data from servers running NameServer, WebCache, WebServer and NetProbes software, and the
30 like. In this example, LogServer software, receives the log file data, and stores the data onto disk. Next, in response to requests for current and historical information from NetView software servers, the LogServer software separates the data into appropriate categories and processes the data to generate statistics. As mentioned above, the data is thus used for monitoring the state of POP servers on the network.

In the present embodiment, various administrator tools, termed AdminTool software are also typically provided. These tools are used by the administrator to configure and administer the network of POP servers. In one example, tools provides tools that “spider” a customer web site and load WebServers and caches with content.

5 In the present embodiment, a single monitoring POP server 50 is contemplated, however, multiple monitoring servers may be provided in alternative embodiments of the present invention.

10 In the present embodiment, POP servers 10-30, and monitoring POP server 50 includes shared software. This set of client and server software programs includes a server that receives a global configuration file that specifies the architecture of the POP servers, and stores it locally.

15 In the present embodiment, in a typical deployment, POP servers will include multiple machines, each with a single purpose (DNS name server, for instance). However, it should be understood that virtually any combination of the software components described above can be installed on a single machine in alternative embodiments of the present invention. Further details regarding the software components will be given below.

20 In the present embodiment, POP servers 10-30 and customer web server 40 and 45 work together to provide static content and dynamic content to a user, such as user 150 coupled to ISP 60. The POP network illustrated in Fig. 1 provides content hosting for static content, such as images, video, software, etc. The POP network also provides for load balancing and traffic management for services, and dynamic content, that cannot be stored. Further detail regarding the process will be described below.

25 In the embodiment illustrated in Fig. 1, it should be noted that the POP server network may simultaneously provide traffic management, as well as content hosting to more than one specific customer. For example, customer1 and customer2 share the POP server network. In one example, POP server network may provide traffic load management by directing traffic to either customer1 web server 40 or customer1 web server 45. At the same time, POP server network may be used to host static content for
30 customer2. Still further POP server may provide both traffic management and content hosting, as will be illustrated and discussed below. Accordingly, the POP server network may be termed a “shared” server network.

Fig. 2 illustrates a block diagram of a typical server machine 300 according to an embodiment of the present invention. Server machine 300 typically

includes a monitor 330, a computer 340, a keyboard 350, an input device 360, and a network interface 370. Computer 340 includes familiar computer components such as a processor 310, and memory storage devices, such as a random access memory (RAM) 320, a disk drive 380, and a system bus 390 interconnecting the above components.

5 Network interface 370 is illustrated coupled to a network 395, or the like

In the present embodiment, a mouse is but one example of an input device 360, also known as a pointing device. Other types of input devices may include trackballs, drawing tablets, microphones (for voice activated input), and the like. In the present embodiment, network interface 370 may be an Ethernet card, a modem, or the
10 like.

RAM 320 and disk drive 380 are examples of tangible media for storage of data, files, log data, computer programs, drivers, embodiments of the herein described methods, and the like. Other types of tangible media include floppy disks, removable hard disks, optical storage media such as CD-ROMS and bar codes, and semiconductor
15 memories such as flash memories, read-only-memories (ROMS), battery-backed memories, and the like.

In a preferred embodiment, Server 300 includes a PC compatible computer having '586 or '686 class based microprocessors, such as the Athlon™ microprocessor from AMD. Further, in the present embodiment, Server 300 utilizes LINUX, and runs
20 software provided by Speedera Networks, Inc., the assignee of the present invention.

Fig. 2 is representative of but one type of system for embodying the present invention. It will be readily apparent to one of ordinary skill in the art that many types of hardware and software configurations are suitable for use in conjunction with the present invention. For example, processors such as the G4 from Motorola, PentiumIII, Pentium4, Itanium from Intel, and the like may be used; further any computer
25 communications bus may be used in alternative embodiments of the present invention. In other embodiments, server 300 may operate under the WindowsNT operating system, may operate under MAC OS from Apple Computer, BeOS from Be, Incorporated, and the like.

30 Figs. 3A-3C illustrate a flow chart according to an embodiment of the present invention. In particular, Figs. 3A-3C illustrate a method for a network of POP servers to service a request from a user. In the flow chart below, reference are made to the system in Fig. 1 for convenience.

Next a user, such as user 150 enters the name of a customer's web site, step 410. In one embodiment, a user can type-in the name of a customer's web site via keyboard, and in another embodiment, the user can select a hyperlink to the customer's web site from a page the user is viewing, or the like. In the example, the customer's web site is "customer1.com."

In response to the user entry, the name of the customer's web site is passed (220) to a local client DNS (LDNS) 170, step 420. As discussed above, domain name servers typically translate textual names, or the like into IP addresses. If the LDNS does not know how to resolve the name, the DNS may route the request to another DNS. In this example, client DNS 170 passes (230) the name to an authoritative NameServer server (ANDS) 110, step 430.

In the present embodiment, NameServer server 110 typically uses the latency and traffic load information associated with the customer's web sites that was determined in step 390, to select an appropriate customer web server to return to the user, step 440. In this example, NameServer server 110 determines that the west coast customer web server 45 is very heavily loaded with requests, but the east coast customer web server 40 has light traffic. Thus, in response to the user's request, NameServer server 110 determines that the user should be directed to customer web server 40. NameServer server 110 then determines the IP address of customer web server 40, step 450, and then sends the IP address to the user, step 460.

In an alternative embodiment of the present embodiment, NameServer server 110 may uses latency information from multiple NetProbe servers such as NetProbe server 670 and 675. Further, NameServer server 110 may uses this information in addition to latencies between the NameServer server and NetProbe servers 670 and 675 in step 440. As an example, NetProbe server 670 reports the latency of customer web server 45 to NameServer 110 because NetProbe server 670 may be the closest NetProbe server; further NetProbe server 675 reports the latency of customer web server 40 to NameServer 110 because NetProbe server 675 may be the closest NetProbe server. In an example, if the reported latencies are the same, NameServer 110 would resolve the domain name to customer web server 45 because the additional latency between NetProbe server 670 and NameServer 110 is smaller than the latency between NetProbe server 675 and NameServer 110. In other words, the sum of the latencies are used to determine how to resolve the IP address. In this example, user 150 receives the IP address, and then sends requests for a web page, or the like, at the received IP address, step 470. The web

page request typically includes static content and dynamic content. In this example, static content may include graphics and text that typically do not change between web page accesses, for example, corporate logos, archival images, documents, and the like. In this example, dynamic content may include graphics and text that do change between web page accesses, for example, stock quotes, search results, live web cameras, and the like.

In the present example, the web page requested from customer web server 40 includes static data, such as HTML text and URL addresses of static content, such as images, and dynamic data, such as stock quotes. In Fig. 1, the static data typically resides on web server 130, and the dynamic data is resides within server 140. As discussed above, this data is also typically mirrored on web server 135 and server 145 in customer web server 45.

In the present embodiment, the customer does not want web server 130 to have the burden of providing static content to many different users. In this embodiment, the POP network provides the content hosting of the static content. Accordingly, the URL addresses of the static content typically include textual domain names specifying the POP server network.

In response to the request from user 150, web server 130 sends the HTML text, or the like, and the URL address of images to user 150, step 480. Additionally, server 140 may process the request from user 150, determine the responsive data dynamically, and send the dynamic data back to user 150, step 490.

In the present embodiment, the user's 150 client software receives the HTML text, dynamic data, URL address of static content, and the like from customer web server 40. Similar to above, the URL address is passed (220) to client DNS (LDNS) 170 for resolution, step 500. In this example, in response, client DNS 170 passes (230) the URL address to NameServer server (ADNS) 110, step 510.

In the present embodiment, NameServer server 110 uses the service metrics information associated with the other POP servers that was determined in step 400, to select an appropriate POP server that includes the static content, step 520. In this example, NameServer server 110 determines that the west coast POP server 10 is very heavily loaded with content requests, but the east coast POP server 30 has light few requests. Thus, in response to the user's request, NameServer server 110 determines that the user should retrieve data from POP server 30. NameServer server 110 then recalls the IP address of POP server 40, step 530, and then sends the IP address to the user, step 540.

Next, user 150 receives the IP address of POP server 30, and then sends requests (250) for the static content from POP server 30, step 550. In response to the request, POP server 30 routes the request to WebCache server 280. Next, WebCache server 280 retrieves the static content from memory, and sends the requested static content back to user 150, step 560.

In an important embodiment of the present invention, the customer's web server may also be considered for providing the static content. In such an embodiment, in step 520, above, the latency and traffic load information for the customer web server may be considered at the same time as the other POP servers . If the customer's web server is lightly loaded compared to the POP servers, the IP of the customer's web server may be retrieved in step 530, and passed to the user in step 540. Accordingly, in step 560, the customer's web server provides the static content back to the user.

Once the client receives the static content, the user can view the HTML text and dynamic data from customer web server 40, as well as view the static content from POP server 30, step 570. In one embodiment, the source of the static content may be specified as being provided at a specific server or a limited set of servers.

Fig. 4 illustrates a flow chart according to another embodiment of the present invention. In particular, Fig. 4 illustrates a more detailed block diagram of step 560, above.

Initially, WebCache server 280 determines whether it has the requested static content in memory, step 600. In one embodiment of the present invention, this step may include determining whether the file requested is resident in memory. If the requested static content is not cached within WebCache server 280, WebCache server 280 sends a request (260) for the static content to WebSever server 290, step 610.

In response, WebServer server 290 determines whether it has the requested static content on disk, or the like, 620. In one embodiment of the present invention, this step may include determining whether the file requested is resident in on a disk or database. If the requested static content is not stored within WebServer server 290, WebServer server 290 determines sends a request (270) for the static content to a customer web server, step 630.

In embodiments of the present invention, the appropriate customer web server selected is also determined based upon the traffic load and latency of the multiple customer web servers to POP server 30. As an example, since POP server 30 is on the east coast and close to customer web server 40. As a result, the request for the static

content is directed to customer web server 40. In an alternative embodiment, the customer web server from which static content is retrieved is predetermined.

In this example, customer web server 40 provides the static content to WebServer server 290, which in turn stores the static content to disk, or the like, step 640.

5 Once the data is stored within WebServer server 290, the static content is copied into the memory of WebCache server 280, step 650. Next, WebCache server 280 returns the static content to the user, step 660.

10 Fig. 5 illustrates a more detailed block diagram according to one embodiment. In particular, Fig. 5 illustrates are more detailed description of the flow of data between software components on the POP network.

15 Fig. 5 includes a reduced number of software components on servers on the POP network, including a NameServer server 700, a WebCache server 710, a WebServer server 720, NetProbes 730, LogServer server 740, and NetView server 750. As illustrated, in the present embodiment, communication between the components is performed using a packetized format such as TCP, UDP, or the like. Fig. 5 also illustrates a customer web site 760.

20 In the present embodiment, the configuration of the POP network is maintained in a single global configuration file. This file is known as the “global configuration” file. In this example, every POP server in the POP network determines configuration information by the copy of the current Speedera configuration file stored locally.

25 In this embodiment, the configuration file includes a list of servers allowed to change the configuration; a list of domains the network is responsible for (i.e. customers’ web sites); list of software components and services available in each POP server; a list of probes that perform latency checks at each POP server; and the like. Other similar types of information is also included.

In the present example, a new configuration file can be pushed to all POP server by using the AdminTools software, mentioned above.

30 The configuration file in the present embodiment, is in a “registry tree” type format. Typically few statistics, status or extended information is kept in the configuration file. Such data is typically stored in server 740, as will be described below. In the present example, the configuration file is kept small to reduce the amount of time required for updating a configuration file across the POP network. Further, a tree-type

format is used to allow the configuration to be easily extended without having to change its structure.

In this embodiment, the configuration file is not necessarily sent to all POP servers in the POP network. Instead, only POP servers including NameServers, NetProbes and LogServers are updated. In the present embodiment, POP servers only including a WebCache or a WebServers are not updated, because WebCache and WebServers access NameServers to find services on the POP network. For example, when a WebCache needs to send information to a log server, it will make a request: “log.name.speedera.com.” In response, a request is generated to a NameServer and in response the NameServer returns the appropriate IP address. Because the configuration file does not need to be sent to all the POP servers in the network, any configuration file update is performed efficiently.

With this configuration file scheme, to install a new POP server, an administrator uses the AdminTools software tools, to register the unique domain name and the services or software components installed on the new POP server. These data are added to the configuration file. Subsequently, the configuration file is copied across the POP network.

In the present embodiment, as mentioned above, the domain name uniquely identifies the POP server. In one example, the name is “name.pop.speedera.net”, “dp1.westusa.speedera.net”, or the like.

As mentioned above, the NameServer software is embodied as a DNS server that performs name to IP address mapping. In particular, the software receives requests containing textual names and returns numeric IP addresses typically in the form of xxx.xxx.xxx.xxx, associated with those names. In the present embodiment, the software is also responsible for receiving and storing latency, POP server status information, and the like from the NetProbe servers. Other types of data includes the source IP address, the availability of the POP server, the traffic load, and the like are also stored.

In the present embodiment, approximately 10 servers including the NameServer software will operate on the POP network. Typically, the NameServer software is installed onto its own machine to increase the performance of the software. In one embodiment, the DNS servers should be designed to handle upwards of 1000 hits per second.

It is contemplated that the 10 DNS servers will handle all the DNS requests for all users on the Internet that are serviced by the POP network. For example, users that require the POP network to determine the least loaded customer web site, and users that require content stored on the POP network both require services from the DNS servers. In alternative embodiments, a greater number of lesser number of DNS servers may be used according to specific application.

In this embodiment, in operation, the NameServer replies to DNS queries and does not return any other information. Specifically, to generate a reply to a DNS query, the DNS server may use the following information: the configuration file to determine what machines have the content corresponding to the request; a table containing latency information for each POP to each client DNS (traffic load); a table containing service metrics (load/availability) information for each POP server, and the like.

In this embodiment, the data received from NetProbe servers comprise UDP traffic. Further the NameServer software transmits log information to the LogServer software, also via UDP traffic. In other embodiments, other schemes for transferring packetized data is contemplated.

The NameServer receives the service and latency information data from the NetProbe servers. In addition, in this embodiment, timestamp is also received so the Name Server software knows how stale the performance information is. The NameServer software stores this information to disk, to RAM, or the like. In this embodiment, the NameServer data is typically stored on disk to reduce the effects of a crash or a reboot. In this way, the latency and probe information is maintained. Further, in some cases, data such as persistence information, should be maintained even if the system is rebooted.

In this embodiment, the NameServers log the requests they get and the replies they give, i.e. IP addresses. In operation, at regular intervals (specified by the global configuration file), the NameServer sends its latest statistics (e.g. number of hits it has taken per domain name, etc.) to the various NetProbe servers and appropriate log analysis servers via UDP. In the global configuration file, log analysis servers are assigned based on the domain name served. In the present embodiment, the log analysis is a distributed process.

In the present embodiment, the DNS software is proprietary and supports all features of BIND.

The WebCache software is typically embodied as a caching web server. As described above, the WebCache server accepts HTTP or URL requests and checks memory for the requested content. As discussed in the flow diagram above, the content is typically static content from a customer's web page, or the like. If the content is not in the memory, the WebCache server contacts a WebServer server that may have the web page stored using HTTP.

In one embodiment, WebCache is a customized caching server. In other embodiments, WebCache may include hardware cache boxes available from companies such as CacheFlow, Inktomi, or the like.

In this example, at a regular fixed intervals, the WebCache server compresses and sends the logs of the content it stores, and sends the logs to log analysis servers. This transfer is typically via UDP traffic. This information provided by the WebCache server is typically used for billing purpose as well as by customers for auditing and log analysis. In the case where a hardware cache box is used, the box must also be configured to send the logs to the log analysis servers.

In the present embodiment, the WebCache software is proprietary but could be integrated with off the shelf caching products.

In the present embodiment, when a WebCache server cannot provide the content requested, it forwards the request to a server running WebServer software it believes has the content. As discussed above, the WebServer software processes the request by looking on its local disk for the content. If the content exists on the local disk, the WebServer will pass the data to the WebCache. Further, if the content is not on the disk, the WebServer contacts the customer's web site to retrieve the content.

In this embodiment, the WebServer typically does not provide log data. Instead, the WebCache provides the cache miss information to the log analysis servers. WebServer, along with many of the other software modules do report error conditions to the log server.

In the present embodiment, the WebServer and WebCache may exist as one program. In alternative embodiments, these pieces of software may be separate and even reside on different machines.

The NetProbes software performs service availability/metric and latency probes and sends the results via UDP to NameServer servers. There are two distinct types of probes in the present embodiment, service probes and latency probes.

In this example, service probes determine service availability and load (metrics) for each content delivery machine in the network. Content delivery machines includes the POP servers including WebCache and WebServer modules, and may also include the customer's web site(s). Service probes monitor data such as HTTP total
 5 response time, FTP total response time, and the like.

In the present embodiment, service probes run constantly, and send current metric and availability information to all DNS servers in the POP network. The timing of the probe intervals and configuration of the service probes is typically specified in the global configuration file.

10 Latency probes determine latency from the POP server hosting the NetProbe to the client DNS servers that sent the requests to Speedera DNS servers. The Speedera DNS servers typically drive the latency probes. In particular, when a DNS server determines that it needs latency information from a probe, it sends a request to the probe. In response, the latency probe probes the client DNS server and returns with the
 15 result. In one embodiment, the DNS servers send a number of requests in a single UDP packet to a latency probe.

The NetProbe servers typically do not store the results of the probes, but simply sends them to other servers via UDP. Each piece of probe information has a timestamp of when the probe occurred so the receiving server can determine how stale the
 20 probe information is.

In the present embodiment, LogServer servers receives data from machines running NameServer, NetProbes and WebCache software. The data is typically transmitted to one or more log analysis servers where the data is stored and analyzed. As mentioned above, the data is typically stored by customer name to facilitate billing and
 25 preparation of audit and historical statistics. Additionally, The LogServer software also gets requests for information from the NetView software. Such a request may be made when an administrator, or the like wants to graphically view or view the table of data for current or historical status of a web site (hits, open connections, etc.).

The configuration file will have a list of log servers and the customer
 30 domains they manage, however, LogServers will only receive data for the customer domains they manage, not for other domains.

The NetView software referred to above is typically an HTML user-interface providing customers and administrators to use a web browser to view real-time

and historical information. The NetView software includes a web server and CGI or web server plug-in modules written in Perl or C.

5 In operation, NetView servers contact LogServers or the LogServer database to get the underlying data. The raw data from LogServers and formats and displays it to customers, or the like in the form of tables and/or graphs.

The NetView software also provides an interface that allows a customer to flush content from the POP network when they publish new content to their servers. To perform the action of flushing content, the NetView software executes the AdminTool software that has the ability to flush content from the system.

10 In the present embodiment, the AdminTools software includes tools used to administer the Speedera Network. It contains two main tools, one to push a new configuration file to servers in the network and another to cause content to be flushed from the network.

15 The configuration administration tool sends the global configuration file to servers in the network in a hierarchical fashion. A list of servers will get the configuration file and send it to another list of servers and so on. The list of servers to receive the configuration file and what servers they control is typically specified by the configuration file itself.

20 In this embodiment, the configuration system is written such that it does not allow configuration files to get out of sync, when machines are down or if a socket connection breaks while transmission is in process, or the like. For example, in one embodiment, the administration tool opens sockets to all the master configuration servers, sends the configuration file and waits for confirmation that they and everyone they send the file to receives it and then send a command to tell everyone to switch to use that
25 configuration file.

In this embodiment, the configuration administration tool uses TCP and not UDP because of the increased security. Additionally, password protection and encryption may also be used.

30 The Shared software includes software that will exist on all servers in the network. This includes a configuration daemon that listens for the transmission of a new configuration file, stores it locally and sends a SIGHUP to local processes to notify them the configuration has changed.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. Many changes or modifications are readily envisioned. For example, in some embodiments, steps 460 and 470 are not explicitly performed. Instead, after the DNS server resolves the IP address of the customer web server in step 450, the IP address is automatically used to address the customer web server. Similarly, in some embodiments, steps 540 and 550 are not explicitly performed. Instead, after the DNS server resolves the IP address of the POP server in step 530, the IP address is automatically used to address the POP server. Details of additional embodiments of the present invention may be found in the documents in the attached appendix.

In embodiments of the present invention, the static content stored on the web caches may include movies, audio, images, text, plug-in program data, programs, and the like. Other types of data can be stored within the web caches, for example, streaming audio data, video data, and the like.

Further, the embodiments have discussed providing data across wide area networks, such as the Internet. In other embodiments, embodiments of the above described invention may be applied to other types of networks including wireless networks, local area networks, and the like.

The block diagrams of the architecture and flow charts are grouped for ease of understanding. However it should be understood that combinations of blocks, additions of new blocks, re-arrangement of blocks, and the like are contemplated in alternative embodiments of the present invention. Further, the above names of servers, software modules, and the like are arbitrary, and may be changed.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

WHAT IS CLAIMED IS:

1 1. A method for operating a network of point of presence servers .
2 sharing a hostname, the method comprises:
3 receiving a request from a user for a web page at a first web address, the first
4 web address including the hostname;
5 determining traffic loads of a plurality of customer web servers, each of the
6 customer web servers storing the web page;
7 determining a customer web server from the plurality of customer web servers
8 that is appropriate for the request, the customer web server having a traffic load lower than
9 traffic loads of remaining customer web servers from the plurality of customer web servers;
10 determining an IP address of the customer web server;
11 directing the request from the user to the customer web server; thereafter
12 receiving a request from the user for static content on the web page at a
13 second web address, the second web address including the hostname;
14 determining service metrics of point of presence servers in the network of
15 point of presence servers;
16 determining the point of presence server from the network of point of
17 presence servers that is appropriate for the request, the point of presence server having service
18 metrics better than service metrics of remaining point of presence servers from the network of
19 point of presence servers;
20 retrieving the static content from the point of presence server; and
21 providing the static content to the user.

1 2. The method of claim 1 further comprising determining load of
2 point of presence servers in the network of point of presence servers;
3 wherein determining the point of presence server from the network of point of
4 presence servers that is appropriate for the request, the point of presence server having a
5 latency and a load lower than latency or load of the remaining point of presence servers from
6 the network of point of presence servers.

1 3. The method of claim 1 further comprising:
2 determining whether the point of presence server includes the static content;
3 determining a web server that includes the static content when the point of
4 presence server does not include the static content;

338333 "034343"

5 retrieving the static content from the web server that includes the static
6 content; and
7 storing the static content from the web server in the point of presence server.

1 4. The method of claim 3 wherein determining the web server
2 comprises:
3 determining traffic loads of the plurality of customer web servers, each of the
4 customer web servers storing the static content; and
5 determining another customer web server from the plurality of customer web
6 servers that is appropriate for the request, the another customer web server having a traffic
7 load lower than traffic loads of remaining customer web servers from the plurality of
8 customer web servers.

1 5. The method of claim 4 wherein retrieving the static content from
2 the web server comprises:
3 determining another IP address of the another customer web server; and
4 requesting the static content from the another customer web server at the
5 another IP address.

1 6. The method of claim 1 wherein the network of point of presence
2 servers comprises a domain name server.

1 7. The method of claim 1
2 wherein the request from the user for the web page is transferred from a first
3 domain name server;
4 wherein the network of point of presence servers comprises a second domain
5 name server; and
6 wherein the second domain name server determines the customer web server
7 from the plurality of customer web servers.

1 8. A method for operating a network of point of presence servers
2 comprises:
3 receiving a first request from a client DNS server to resolve a first domain
4 name, the client DNS server receiving a request from a user of a web page address that
5 includes the first domain name;

6 determining load measurements of a plurality of customer web servers, each
 7 of the customer web servers addressable by the first domain name, and each of the customer
 8 web servers configured to service the request from the user;

9 determining a customer web server from the plurality of customer web
 10 servers, the customer web server having load measurements lower than load measurements of
 11 other customer web servers from the plurality of customer web servers;

12 determining an IP address of the customer web server;

13 providing the IP address of the customer web server to the client DNS server;
 14 thereafter

15 receiving a second request from the client DNS server to resolve a second
 16 domain name, the client DNS server receiving a request from the user of a uniform resource
 17 locator that includes the second domain name;

18 determining performance metric measurement of point of presence servers in
 19 the network of point of presence servers, each of the point of presence servers addressable by
 20 the second domain name;

21 determining a point of presence server from the network of point of presence
 22 servers, the point of presence server having performance metric measurements lower than
 23 performance metric measurements of other point of presence servers from the network of
 24 point of presence servers;

25 providing the IP address of the point of presence server to the client DNS
 26 server;

27 retrieving data from the point of presence server in response to the uniform
 28 resource locator; and

29 providing the data to the user.

1 9. The method of claim 8 wherein the load measurements comprise
 2 latency measurements.

1 10. The method of claim 8 wherein the performance metric
 2 measurements comprise load CPU and memory measurements, HTTP response
 3 measurements, FTP response measurements.

1 11. The method of claim 8 wherein retrieving data from the point of
 2 presence server comprises:

3 determining whether the point of presence server includes the data;

4 retrieving data from another customer web server from the plurality of
 5 customer web servers when the point of presence server does not include the data; and
 6 storing the data within the point of presence server.

1 12. The method of claim 11 wherein retrieving data from the other
 2 customer web server comprises:

3 determining the other customer web server from the plurality of customer web
 4 servers, the other customer web server having load measurements lower than load
 5 measurements of remaining customer web servers from the plurality of customer web servers;
 6 and

7 retrieving the data from the other customer web server.

1 13. The method of claim 8 further comprising:

2 receiving a first request from a second client DNS server to resolve a third
 3 domain name, the second client DNS server receiving a request from a second user of a
 4 second web page address that includes the third domain name;

5 determining load measurements of a plurality of second customer web
 6 servers, each of the second customer web servers addressable by the third domain name, and
 7 each of the second customer web servers storing data configured to service the request from
 8 the second user;

9 determining a second customer web server from the plurality of second
 10 customer web servers, the second customer web server having load measurements lower than
 11 load measurements of other second customer web servers from the plurality of second
 12 customer web servers;

13 determining an IP address of the second customer web server; and

14 providing the IP address of the second customer web server to the second
 15 client DNS server.

1 14. The method of claim 13 further comprising:

2 receiving a second request from the second client DNS server to resolve the
 3 second domain name, the second client DNS server receiving a request from the second user
 4 of a second uniform resource locator that includes the second domain name;

5 retrieving a second set of data from the point of presence server in response to
 6 the second uniform resource locator; and

7 providing the second set of data to the user.

1 15. A method for a network of point of presence servers comprises:
2 receiving at a first point of presence server a first request from a first client
3 DNS server to resolve a first domain name, the first request from the first client DNS server
4 determined in response to a first uniform resource locator entered by a first user, the first
5 uniform resource locator comprising the first domain name;
6 receiving at a second point of presence server a first request from a second
7 client DNS server to resolve the first domain name, the first request from the second client
8 DNS server determined in response to the first uniform resource locator entered by a second
9 user, the first uniform resource locator comprising the first domain name;
10 determining at the first point of presence server traffic measurements of a
11 plurality of customer web servers, each of the customer web servers addressable by the first
12 domain name, and each of the customer web servers storing data associated with the first
13 uniform resource locator;
14 determining at the first point of presence server a first customer web server
15 from the plurality of customer web servers, the first customer web server having traffic
16 measurements lower than load measurements of other customer web servers from the
17 plurality of customer web servers;
18 determining at the second point of presence server the first customer web
19 server from the plurality of customer web servers, the first customer web server having
20 traffic measurements lower than load measurements of other customer web servers from the
21 plurality of customer web servers;
22 determining at the first point of presence server an IP address of the first
23 customer web server;
24 determining at the second point of presence server an IP address of the first
25 customer web server;
26 providing from the first point of presence server the IP address of the first
27 customer web server to the first client DNS server and to the second client DNS server;
28 providing from the second point of presence server the IP address of the first
29 customer web server to the first client DNS server and to the second client DNS server;
30 thereafter
31 receiving at the first point of presence server a second request from the first
32 client DNS server to resolve a second domain name, the second request from the first client
33 DNS server determined in response to a second uniform resource locator comprising the
34 second domain name, the second uniform resource locator from the first customer web server;

35 receiving at the second point of presence server a second request from the
 36 second client DNS server to resolve the second domain name, the second request from the
 37 second client DNS server determined in response to the second uniform resource locator
 38 comprising the second domain name, the second uniform resource locator from the first
 39 customer web server;

40 determining at the first point of presence server performance metric
 41 measurement between the first point of presence server and other point of presence servers in
 42 the network of point of presence servers, each of the point of presence servers addressable by
 43 the second domain name;

44 determining at the second point of presence server performance metric
 45 measurement between the second point of presence server and other point of presence servers
 46 in the network of point of presence servers;

47 determining at the first point of presence server performance metric
 48 measurement between the second point of presence server and other point of presence servers
 49 in the network of point of presence servers;

50 determining at the first point of presence server a third point of presence
 51 server from the network of point of presence servers, the third point of presence server having
 52 performance metric measurements lower than performance metric measurements with regards
 53 to the first point of presence server;

54 determining at the second point of presence server a fourth point of presence
 55 server from the network of point of presence servers, the fourth point of presence server
 56 having performance metric measurements lower than performance metric measurements with
 57 regards to the second point of presence server;

58 retrieving data from a third point of presence server;
 59 retrieving the data from the fourth point of presence server;
 60 providing the data to the first user; and
 61 providing the data to the second user.

1 16. The method of claim 15 wherein the traffic measurements
 2 comprise latency measurements.

1 17. The method of claim 15 wherein the performance metric
 2 measurements comprise CPU load measurements, HTTP response measurements, FTP
 3 response measurements.

1 18. The method of claim 15 wherein retrieving data from the third
2 point of presence server comprises:
3 determining whether the third point of presence server includes the data;
4 retrieving data from another customer web server from the plurality of
5 customer web servers when the third point of presence server does not include the data; and
6 storing the data within the third point of presence server.

1 19. The method of claim 18 wherein retrieving data from the other
2 customer web server comprises:
3 determining the other customer web server from the plurality of customer web
4 servers, the other customer web server having load measurements lower than load
5 measurements of remaining customer web servers from the plurality of customer web servers;
6 and
7 retrieving the data from the other customer web server.

1 20. The method of claim 15 wherein the first domain name and the
2 second domain name are the same.

METHOD FOR OPERATING AN INTEGRATED
POINT OF PRESENCE SERVER NETWORK

ABSTRACT OF THE DISCLOSURE

5

A method for operating a network of point of presence servers sharing a hostname includes receiving a request from a user for a web page at a first web address, the first web address including the hostname, determining traffic loads of a plurality of customer web servers, each of the customer web servers storing the web page, determining a customer web server from the plurality of customer web servers that is appropriate for the request, the customer web server having a traffic load lower than traffic loads of remaining customer web servers from the plurality of customer web servers, determining an IP address of the customer web server, directing the request from the user to the customer web server, thereafter receiving a request from the user for static content on the web page at a second web address, the second web address including the hostname, determining service metrics of point of presence servers in the network of point of presence servers, determining the point of presence server from the network of point of presence servers that is appropriate for the request, the point of presence server having service metrics more appropriate than service metrics of remaining point of presence servers from the network of point of presence servers, retrieving the static content from the point of presence server, and providing the static content to the user.

PA 3088896 v1

© 2000 Network World

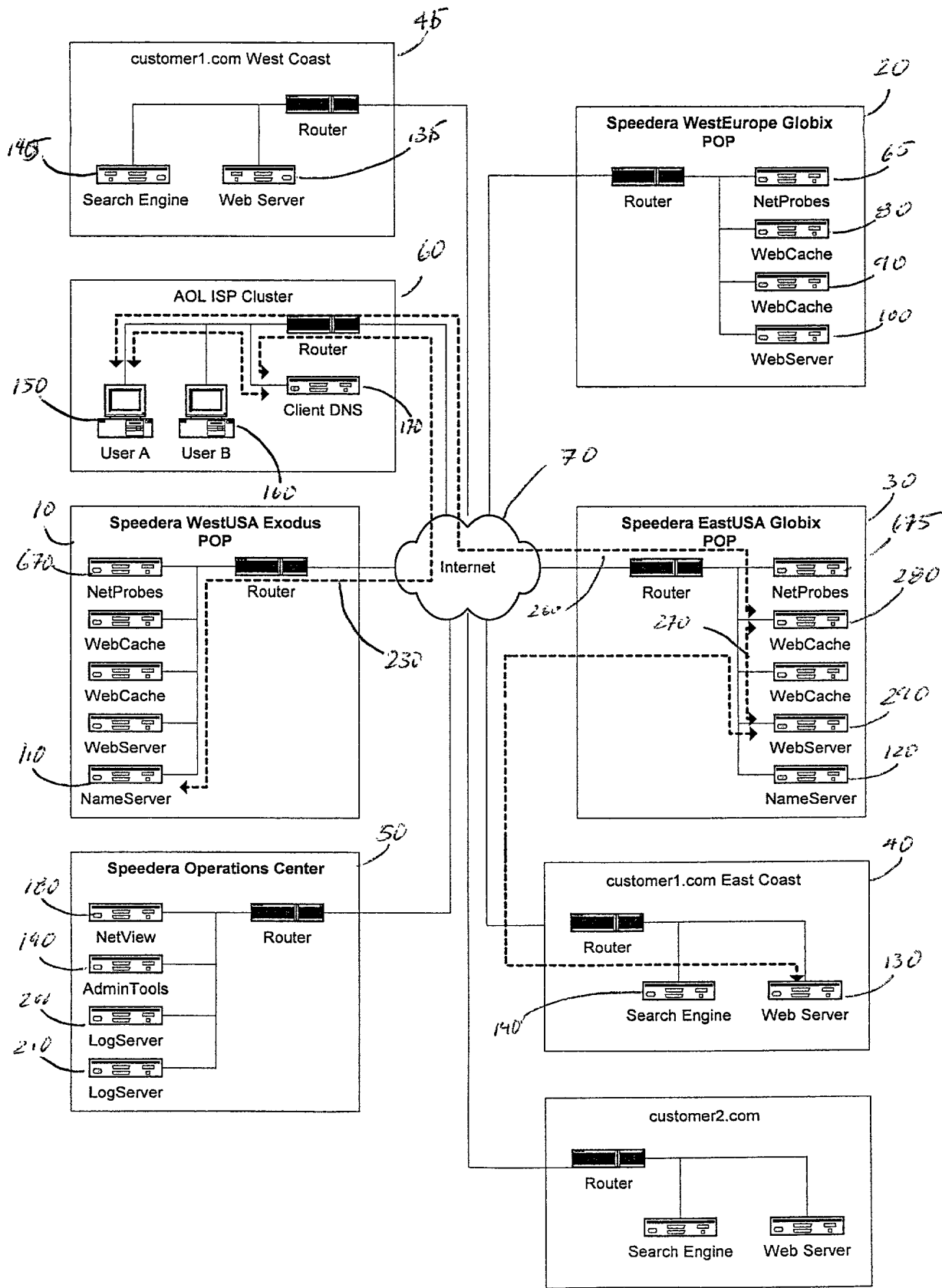


FIG. 1

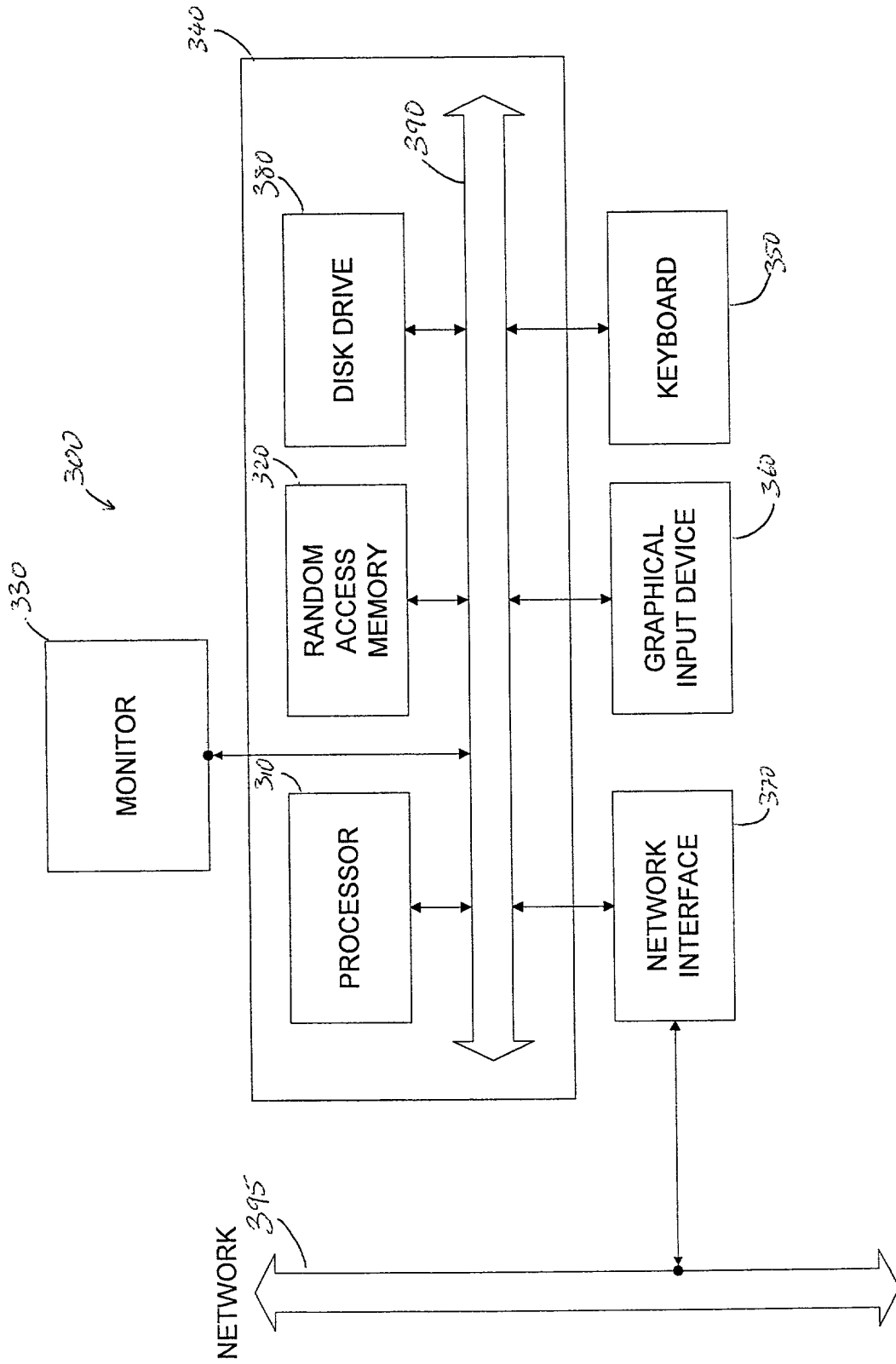


FIG. 2

390 400 410 420 430 440 450

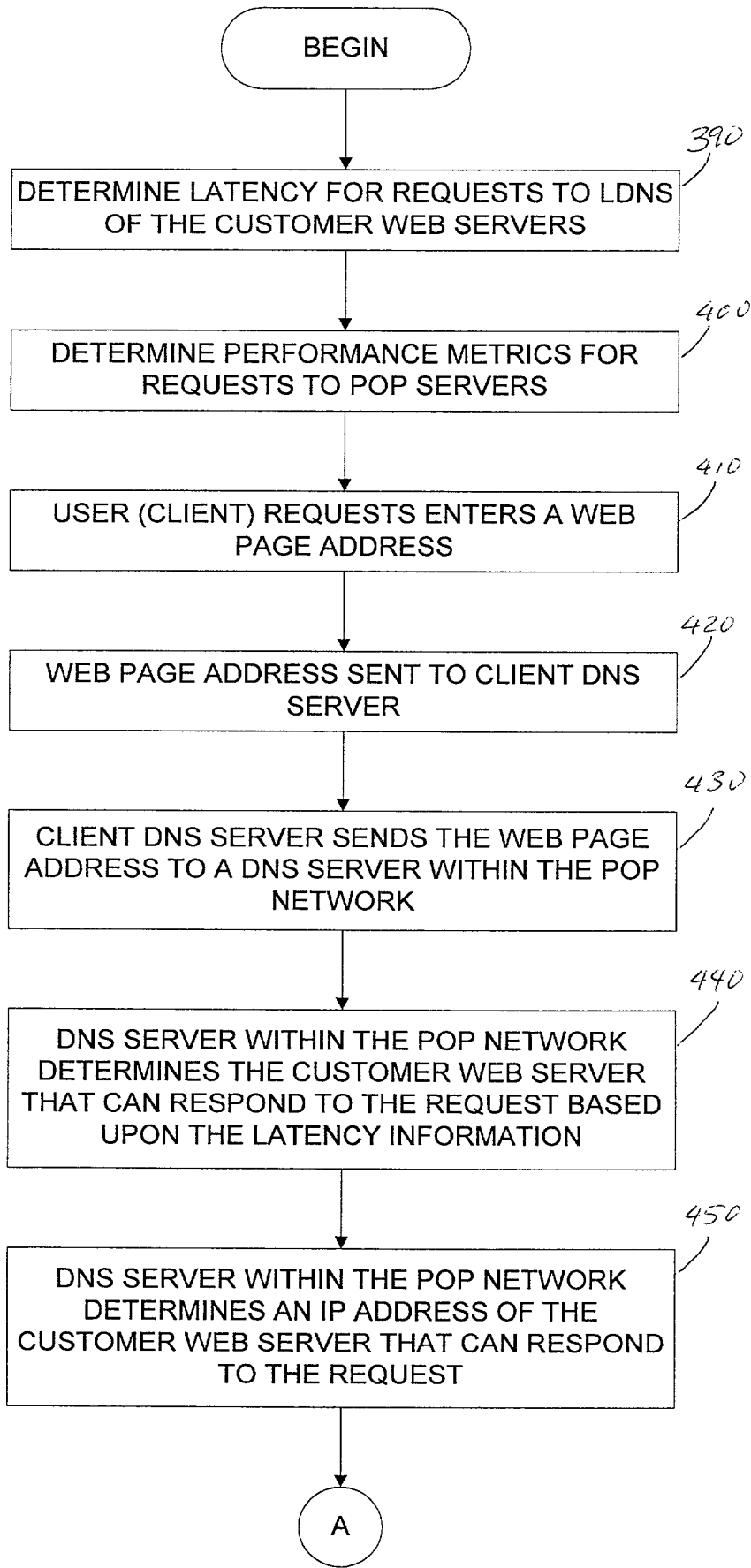


FIG. 3A

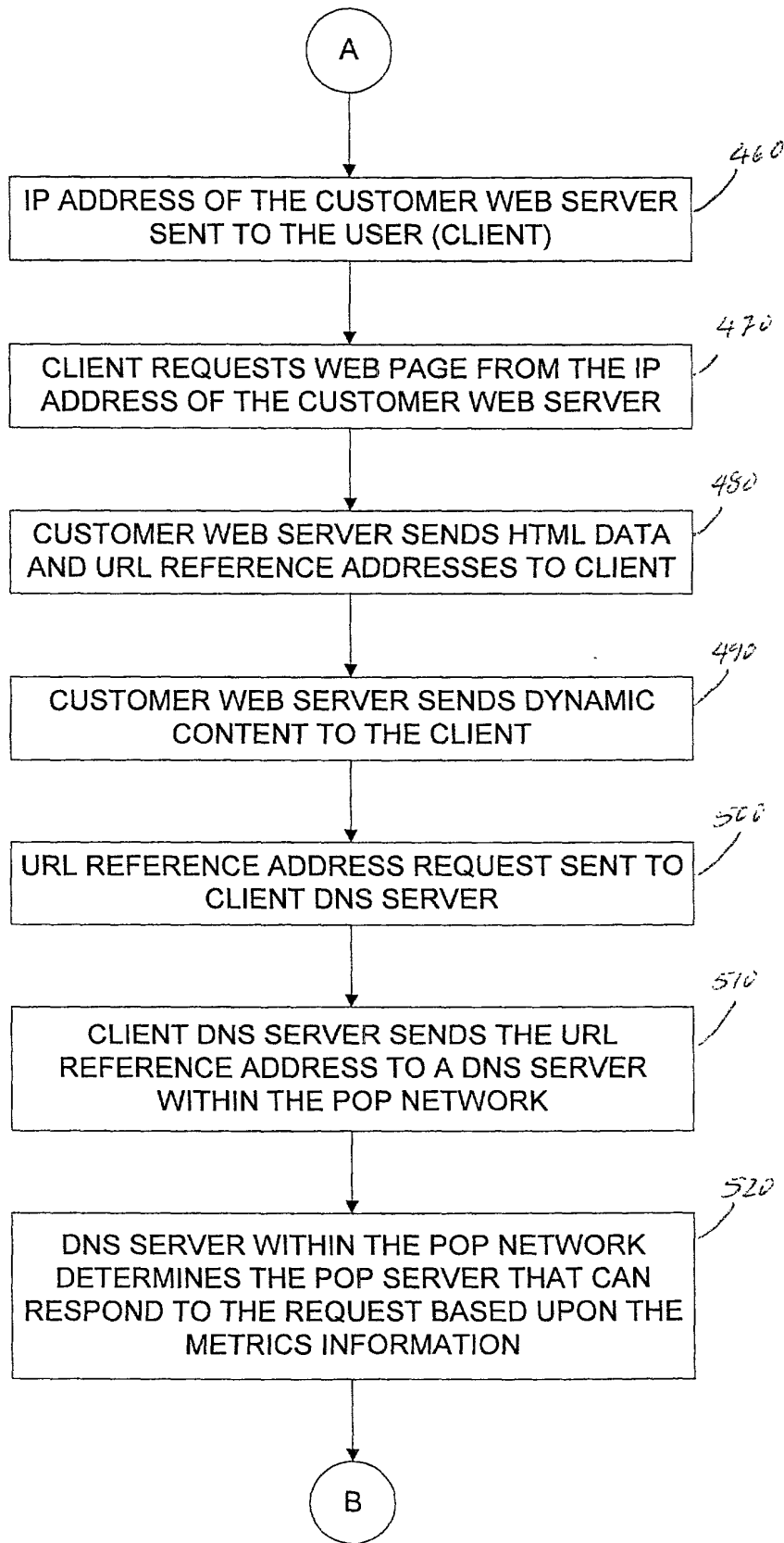


FIG. 3B

00000000000000000000

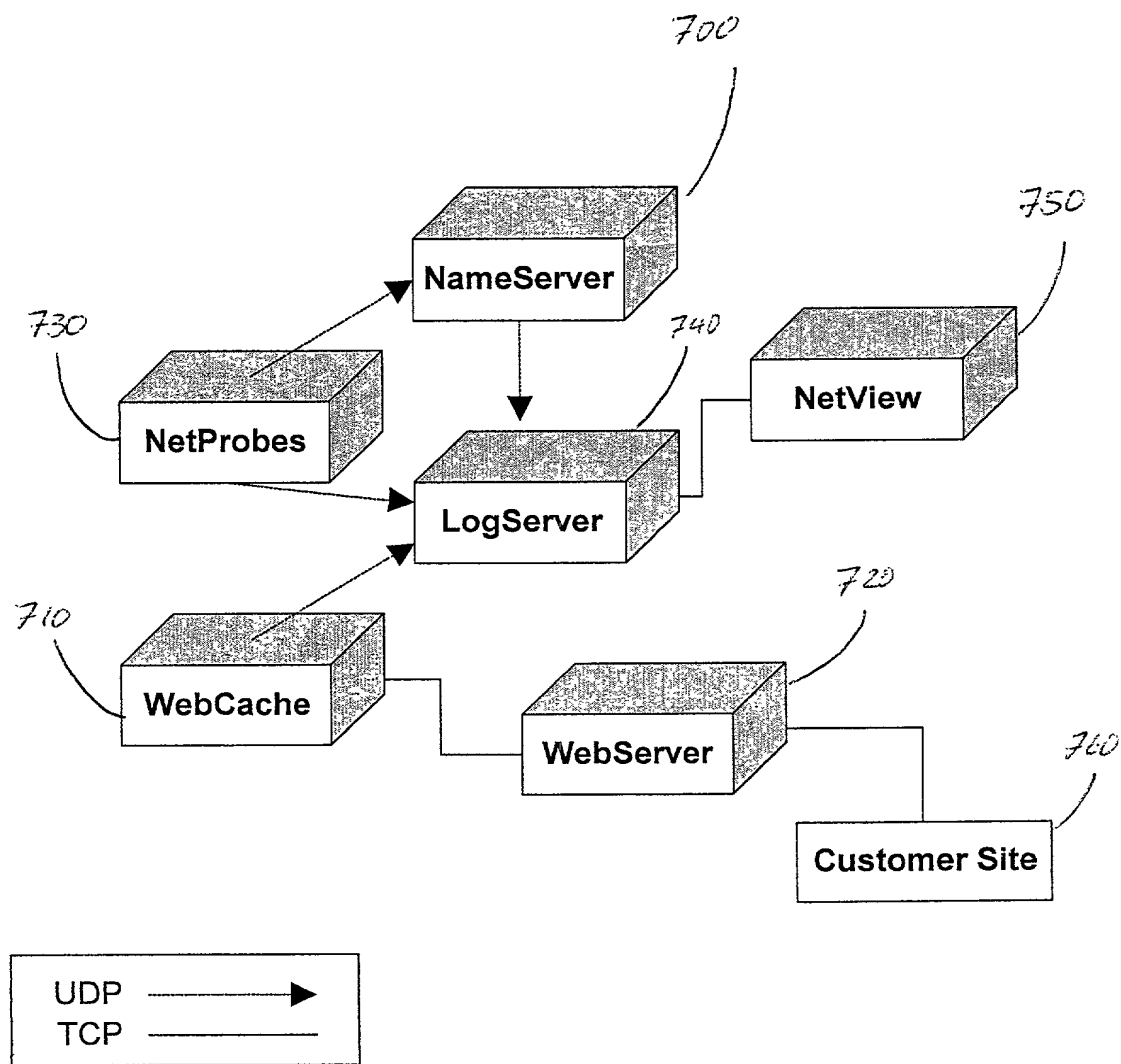


FIG. 5

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **METHOD FOR OPERATING AN INTEGRATED POINT OF PRESENCE SERVER NETWORK** the specification of which X is attached hereto or was filed on _____ as Application No. _____ and was amended on _____ (if applicable).

I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application No.	Filing Date
60/166,906	November 22, 1999

I claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application No.	Date of Filing	Status

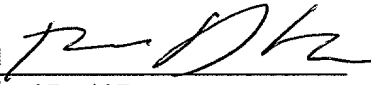
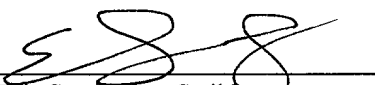

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Stephen Y. Pang, Reg. No. 38,575
 Sujit B. Kotwal, Reg. No. 43,336
 Richard T. Ogawa, Reg. No. 37,692

Send Correspondence to: Stephen Y. Pang TOWNSEND and TOWNSEND and CREW LLP Two Embarcadero Center, 8th Floor San Francisco, California 94111-3834	Direct Telephone Calls to: (Name, Reg. No., Telephone No.) Name: Stephen Y. Pang Reg. No.: 38,575 Telephone: 650-326-2400
--	---

Full Name of Inventor 1:	Last Name: DAY	First Name: RICHARD	Middle Name or Initial: David	
Residence & Citizenship:	City: Mountain View	State/Foreign Country: California	Country of Citizenship: United States	
Post Office Address:	Post Office Address: 912 Rich Avenue, #3	City: Mountain View	State/Country: California	Postal Code: 94040
Full Name of Inventor 2:	Last Name: SWILDENS	First Name: ERIC	Middle Name or Initial: Sven-Johan	
Residence & Citizenship:	City: Mountain View	State/Foreign Country: California	Country of Citizenship: United States	
Post Office Address:	Post Office Address: 723 Tiana Lane	City: Mountain View	State/Country: California	Postal Code: 94041
Full Name of Inventor 3:	Last Name: GUPTA	First Name: AJIT	Middle Name or Initial: KUMAR	
Residence & Citizenship:	City: Fremont	State/Foreign Country: California	Country of Citizenship: India	
Post Office Address:	Post Office Address: 33635 Quail Run Road	City: Fremont	State/Country: California	Postal Code: 94555

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1 	Signature of Inventor 2 	Signature of Inventor 3 
Richard David Day	Eric Sven-Johan Swildens	Ajit Gupta
Date 8/18/00	Date 8/18/00	Date 8/18/00